JCTC Journal of Chemical Theory and Computation

Article

Cite This: J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

pubs.acs.org/JCTC

Extreme Scalability of DFT-Based QM/MM MD Simulations Using MiMiC

Viacheslav Bolnykh,[*,†,‡,¶] Jógvan Magnus Haugaard Olsen,[*,§] Simone Meloni,[∥] Martin P. Bircher,[⊥] Emiliano Ippoliti,[¶] Paolo Carloni,[*,¶,#,†] and Ursula Rothlisberger[*,⊥]

[†]Department of Physics, RWTH Aachen University, 52056 Aachen, Germany

[‡]CaSToRC, The Cyprus Institute, 2121 Aglantzia, Nicosia, Cyprus

[¶]Institute for Advanced Simulation (IAS-5) and Institute of Neuroscience and Medicine (INM-9), Forschungszentrum Jülich, 52425 Jülich, Germany

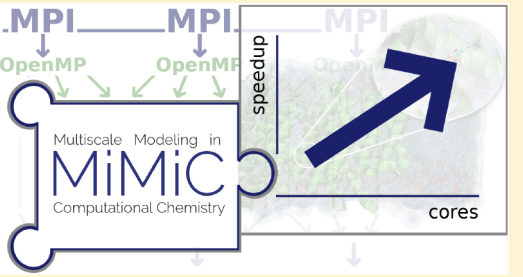
[§]Hylleraas Centre for Quantum Molecular Sciences, Department of Chemistry, UiT The Arctic University of Norway, N-9037 Tromsø, Norway

[∥]Dipartimento di Scienze Chimiche e Farmaceutiche (DipSCF), Università degli Studi di Ferrara (Unife), Via Luigi Borsari 46, I-44121 Ferrara, Italy

[⊥]Laboratory of Computational Chemistry and Biochemistry, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland

[#]Institute for Neuroscience and Medicine (INM-11), Forschungszentrum Jülich, 52425 Jülich, Germany

ABSTRACT: We present a highly scalable DFT-based QM/MM implementation developed within MiMiC, a recently introduced multiscale modeling framework that uses a loose-coupling strategy in conjunction with a multiple-program multiple-data (MPMD) approach. The computation of electrostatic QM/MM interactions is parallelized exploiting both distributed- and shared-memory strategies. Here, we use the efficient CPMD and GROMACS programs as QM and MM engines, respectively. The scalability is demonstrated through large-scale benchmark simulations of realistic biomolecular systems employing non-hybrid and hybrid GGA exchange−correlation functionals. We show that the loose-coupling strategy adopted in MiMiC, with its inherent high flexibility, does not carry any significant computational overhead compared to a tight-coupling scheme. Furthermore, we demonstrate that the adopted parallelization strategy enables scaling up to 13,000 CPU cores with efficiency above 70%, thus making DFT-based QM/MM MD simulations using hybrid functionals at the nanosecond scale accessible.



INTRODUCTION

The scale spanned by (bio)chemical processes ranges from the decomposition of small organic molecules due to attosecond laser pulses to the folding of proteins comprising hundreds of thousands of atoms over several micro- to milliseconds. Exploring the dynamics of those processes requires methods of varying precision and computational cost, where usually a trade-off between the former and the latter has to be found in order to describe chemically relevant time scales and system sizes. In particular, while the system sizes that can be afforded with accurate *first-principles* methods are constantly growing, the requirement of sampling a sufficiently long time scale often limits the system size to a few hundreds of atoms in Kohn−Sham density functional theory (KS-DFT) based molecular dynamics (MD) and to less than a few dozen for accurate wave function-based methods.

One way in which this computational bottleneck can be overcome is the use of multiscale approaches that bridge size scales by combining methods of different accuracy and computational efficiency. The parts of the system that are relevant to a chemical process, in which chemical bonds are broken and formed, are often small with respect to the overall system size. Thus, they can be described using methods of higher accuracy that would not be viable for a description of the entire system. On the other hand, it is usually sufficient for the large remainder of the system to be described by less precise but computationally much more efficient models. The combination of such methods, reducing the computational cost per time step of the calculation, ultimately allows one to cover the time scale necessary to investigate the process considered. Highly popular for, but not limited to, biochemical applications, the quantum mechanics/molecular mechanics (QM/MM) approach,[1−3] allows, for example, a chemically relevant part of a protein to be simulated using quantum mechanics (QM) methods, while the rest of the system can be efficiently described at the molecular mechanics (MM) level with sufficient accuracy. As such, the QM/MM method has been implemented in various

Received: May 2, 2019
Published: September 9, 2019

DOI: 10.1021/acs.jctc.9b00424
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

program-specific contexts in which QM, MM, and the interface between the two subsystems is hard-coded in the same program (tight-coupling).

MiMiC[4] (multiscale modeling in computational chemistry) is a recently introduced framework for multiscale modeling based on a loose-coupling paradigm. Loose coupling allows for maximum flexibility by allowing the framework to be interfaced with different programs with only minimal modifications. However, such an approach should not compromise efficiency. Alternatively, a tight-coupling strategy is limited to the program it has been designed for, with the immediate advantage of making it easy to minimize the computational overhead of the framework for the very programs it interfaces. This, however, comes at the cost of a loss of flexibility. MiMiC aims at uniting the flexibility of a loose-coupling scheme with the efficiency of a tight-coupling approach by virtue of highly efficient data exchange, thus providing a highly efficient loose-coupling framework. Preliminary benchmarks presented in ref 4 demonstrated that our QM/MM implementation using MiMiC can greatly outperform existing, tight-coupling QM/MM implementations.[8]

Here, we present the parallelization schemes at the heart of the MiMiC framework and use a variety of systems to demonstrate that a loose coupling can be exploited while still achieving extreme scalability in DFT-based QM/MM MD simulations. Using systems with different, representative sizes of QM and MM regions, we provide an extensive benchmark and performance study of our QM/MM implementation using MiMiC to interface the CPMD[5] program, a plane wave/pseudopotential implementation of DFT, and the classical MD package GROMACS.[6,7] By using the CPMD program as the main MD driver, both Born−Oppenheimer (BO) and Car−Parrinello (CP) QM/MM MD simulations can be performed. Among others, we present scaling benchmarks using a large membrane protein system (ClC-ec1, a Cl/H+ antiporter). We envisage that, thanks to the extreme scalability of our QM/MM implementation, it will be possible to reach long simulation times also for systems as big as ClC-ec1 when using a hybrid exchange−correlation functional. This will allow, among others, well-converged enhanced-sampling QM/MM MD simulations of very large (bio)molecular systems at a more accurate QM level.

The text is organized as follows: We first present the details of the parallelization schemes implemented in MiMiC, followed by a description of the QM/MM MD workflow and the interprogram communication mechanisms. We then provide a description of the test systems used in the benchmarks. Following this, we present the results of the benchmarks showcasing the parallel performance of the MiMiC-based QM/MM implementation for up to several thousand CPU cores. We show that, at a large number of cores, parallel efficiency is mainly exhausted due to serial parts of the main MD driver in CPMD. This demonstrates that the loose-coupling scheme adopted in MiMiC does not lead to any significant overhead associated with communication or interfacing.

PARALLELIZATION SCHEMES

Here we present the parallelization schemes implemented in MiMiC which utilize distributed memory architectures (via MPI), shared-memory approaches (based on OpenMP), or a combination of both. To this end, we only include key equations of our electrostatic embedding approach and detail their parallel implementation within MiMiC. Readers are referred to ref 4 for a thorough description of the MiMiC framework and the adopted QM/MM methodology.

The energy of the whole system is split into three contributions (as usual for additive QM/MM schemes): the energy of the QM subsystem, the energy of the MM subsystem, and the QM/MM interaction energy. The QM subsystem is described with plane-wave/pseudopotential-based KS-DFT, while the MM subsystem is represented by an MM force field. In the present work, the CPMD program serves as the main MD driver and also computes the QM contributions. GROMACS, in turn, provides the MM contributions as well as bonded interactions (only present when bonds cross the QM−MM interface) and nonbonded van der Waals interactions between the QM and MM subsystems. Through the loose coupling enabled by the MiMiC framework, which allows the external programs to run simultaneously and independently of each other (with periodic data exchange detailed in the Interprogram Communication section below), these contributions can be computed utilizing the respective parallelization schemes implemented in CPMD and GROMACS.

The electrostatic QM−MM interactions, which include interactions between the MM charges (or multipoles) and the QM nuclei and electrons, are computed by MiMiC. This involves the electronic density of the QM subsystem, which is mapped on a grid, typically of the order of 10^6 points, and thus requires special attention, in particular with respect to computational cost and parallel scaling. Evaluating the full electrostatic part of the nonbonded energy exactly is not only computationally expensive but also exhibits poor scalability with the size of the system, which can be of the order of 10^5 atoms or larger, since the full integral over the electronic grid has to be calculated for each of the MM charges (see below). To overcome these limitations, we have formulated and implemented a generalized version of the electrostatic coupling scheme by Laio et al.,[8] which was shown to yield the same accuracy as an exact electrostatic treatment at a substantially reduced cost.[4]

Within the generalized electrostatic coupling scheme, the MM atoms are grouped into two domains according to their distance from the QM subsystem. The MM atoms that are within a specified distance from the QM subsystem (the short-range domain) interact directly with the QM electron density and nuclei, whereas the remaining MM atoms (the long-range domain) interact through a multipole expansion of the electrostatic potential created by the QM electrons and nuclei. In multi-index notation[4] the full electrostatic QM−MM interaction energy can thus be written as a sum of short- and long-range contributions

$$
\begin{aligned}
V^{\mathrm{es}}_{\mathrm{QM/MM}}[\rho^{\mathrm{QM}}](\mathbf{R}^{\mathrm{QM}}, \mathbf{R}^{\mathrm{MM}}) = & \sum_{i=1}^{N^{\mathrm{sr}}_{\mathrm{MM}}} \sum_{|\alpha|=0}^{A_i} \frac{(-1)^{|\alpha|}}{\alpha!} M_i^{[\alpha]} \\
& \times \left(\int T^{[\alpha]}_{\mathrm{mod}}(\mathbf{R}_i^{\mathrm{MM}}, \mathbf{r}) \rho^{\mathrm{QM}}(\mathbf{r}) \mathrm{d}\mathbf{r} + \sum_{j=1}^{N_{\mathrm{QM}}} T^{[\alpha]}_{\mathrm{mod}}(\mathbf{R}_i^{\mathrm{MM}}, \mathbf{R}_j^{\mathrm{QM}}) Z_j^{\mathrm{QM}} \right) \\
& + \sum_{i=1}^{N^{\mathrm{lr}}_{\mathrm{MM}}} \sum_{|\alpha|=0}^{A_i} \sum_{|\beta|=0}^{B_{\mathrm{QM}}} \frac{(-1)^{|\alpha+\beta|}}{\alpha!\beta!} M_i^{[\alpha]} T^{[\alpha+\beta]}(\mathbf{R}_i^{\mathrm{QM}}, \bar{\mathbf{R}}^{\mathrm{QM}}) \\
& \times M^{[\beta]}_{\mathrm{QM}}[\rho^{\mathrm{QM}}](\mathbf{R}^{\mathrm{QM}})
\end{aligned} \tag{1}
$$

where $N^{\mathrm{sr}}_{\mathrm{MM}}$ and $N^{\mathrm{lr}}_{\mathrm{MM}}$ are the number of MM atoms in the short- and long-range domains, respectively, $\mathbf{R}^{\mathrm{MM}}$ and $\mathbf{R}^{\mathrm{QM}}$ are the positions of MM atoms and QM nuclei, respectively, $M_i^{[\alpha]}$ is a component of the multipole associated with ith MM atom,

DOI: 10.1021/acs.jctc.9b00424
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

$\rho^{\mathrm{QM}}(\mathbf{r})$ is the QM electronic density, N_{QM} is the number of QM nuclei, Z_j^{QM} denotes the core charge of the jth QM nucleus, and $M_{\mathrm{QM}}^{[\beta]}[\rho^{\mathrm{QM}}](\mathbf{R}^{\mathrm{QM}})$ is a multipole component of the QM charge density. Eq 1 also involves the unmodified and modified interaction tensors which are generally defined as

$$T^{[\alpha]}(\mathbf{R}_a, \mathbf{R}_b) = \partial^{\alpha}_{\mathbf{R}_b} \frac{1}{|\mathbf{R}_b - \mathbf{R}_a|}$$

and

$$T^{[\alpha]}_{\mathrm{mod}}(\mathbf{R}_a, \mathbf{R}_b) = \partial^{\alpha}_{\mathbf{R}_b} \frac{r_{c,a}^4 - |\mathbf{R}_b - \mathbf{R}_a|^4}{r_{c,a}^5 - |\mathbf{R}_b - \mathbf{R}_a|^5},$$

respectively, where

$$\partial^{\alpha}_{\mathbf{R}_b} = \frac{\partial^{|\alpha|}}{\partial R_{b,x}^{\alpha_x} \partial R_{b,y}^{\alpha_y} \partial R_{b,z}^{\alpha_z}},$$

and $r_{c,a}$ in the latter is the covalent radius of the ath MM atom. The modified interaction tensor is introduced in order to mitigate divergences at short distances. This prevents electron spill-out which is pervasive in electrostatic embedding QM/MM due to the lack of Pauli repulsion between the QM and MM subsystems. The modified form of the interaction potential was introduced and validated in ref 8.

The most computationally intensive part of the electrostatic QM−MM interaction energy calculation comes from the interaction between the MM atoms in the short-range domain and the QM electrons. This involves $N_{\mathrm{MM}}^{\mathrm{sr}}$ integrals over the electronic density, performed on a dense grid. The basic parallelization strategy consists in distributing subsets of the (real-space) grid planes among MPI processes,[9] each of which computes a partial contribution of the energy. We note that while we in this work make use of the plane-wave implementation of the CPMD program, MiMiC is immediately able to support any other type of basis set that allows for a mapping onto real-space grids, including Gaussian-type orbital (GTO) basis sets. Extensions to provide intrinsic support for nongrid-based methods, such as those using GTO basis sets, are possible. The computation of the energy stemming from interactions between MM atoms in the short-range domain and the QM nuclei as well as interactions involving MM atoms in the long-range domain is parallelized by distributing subsets of (short- and long-range) MM atoms to the MPI processes. The full energy is then obtained through global reduction.

There are several approaches to grid decomposition. Here we use a slab decomposition approach where subsets of contiguous real-space grid planes are distributed among the MPI processes. This approach works well as long as the number of MPI processes does not exceed the number of grid planes along the decomposition axis. The scalability can be extended beyond this limit by the use of one or both of two complementary approaches, task grouping of MPI processes and multithreading using OpenMP. To compute the electronic part of the energy, the task group approach distributes the short-range MM atoms among the groups and distributes the grid planes between the MPI processes of each group (see illustration in Figure 1). The OpenMP parallelism further decomposes the grid slabs assigned to a given MPI process into pencils. Instead, the nuclear part is parallelized by distributing the short-range MM atoms across all threads assigned to each MPI process within every group. The total electrostatic QM−MM interaction energy contributions are obtained through reduction operations within each task group as well as across task groups. It is worth noting that the parallelization approach described above has equivalents in the parallelization strategy adopted in CPMD (low-level distribution over planes,[9] task grouping[10−12]—known as *CP groups*[12]—and multithreading[13]).

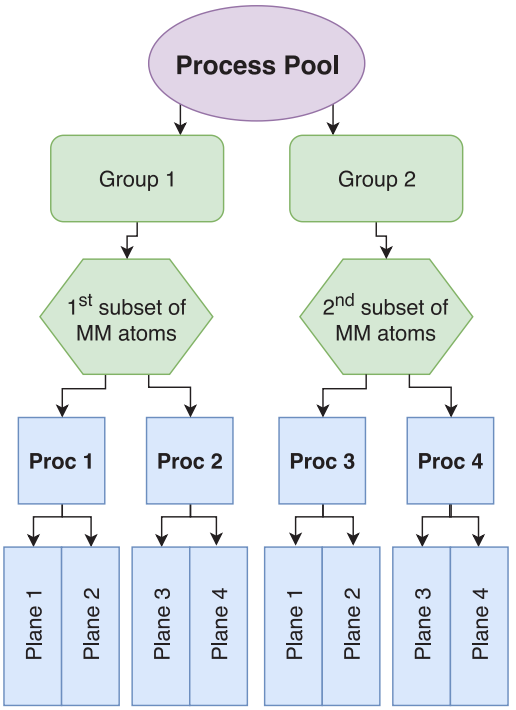


Figure 1. Illustration of the task group parallelization strategy used in the MiMiC-based QM/MM. Here, the MPI processes are divided into groups, each receiving a subset of MM atoms (short- and long-range) and each MPI process of a group receiving a subset of grid planes.

The long-range part of the electrostatic QM−MM interaction energy involves the computation of multipoles associated with the QM charge density defined as

$$M_{\mathrm{QM}}^{[\beta]}[\rho^{\mathrm{QM}}](\mathbf{R}^{\mathrm{QM}}) = \int \rho^{\mathrm{QM}}(\mathbf{r})(\mathbf{r} - \bar{\mathbf{R}}^{\mathrm{QM}})^{\beta} \mathrm{d}\mathbf{r} + \sum_{j=1}^{N_{\mathrm{QM}}} Z_j^{\mathrm{QM}}(\mathbf{R}_j^{\mathrm{QM}} - \bar{\mathbf{R}}^{\mathrm{QM}})^{\beta} \tag{2}$$

where $\bar{\mathbf{R}}^{\mathrm{QM}}$ is the origin of the multipole expansion, for which we use the centroid of the QM subsystem, i.e., $\bar{\mathbf{R}}^{\mathrm{QM}} = \sum_{j=1}^{N_{\mathrm{QM}}} \mathbf{R}_j^{\mathrm{QM}}/N_{\mathrm{QM}}$. The parallelization of the computation of the electronic part of the multipoles is done exclusively by slab decomposition of the real-space grid across MPI processes and the associated OpenMP threads. The nuclear multipoles are computed in a purely serial fashion as this part is a negligible contribution of the overall computational cost of the calculation.

The MM charges (or multipoles) are the source of a (polarizing) electrostatic potential that is added to the external KS potential. The potential is obtained as the functional derivative of the electrostatic QM−MM interaction energy (eq 1) with respect to the electronic density that yields

$$v_{\mathrm{QM/MM}}^{\mathrm{es,sr}}(\mathbf{r}) = \sum_{i=1}^{N_{\mathrm{MM}}^{\mathrm{sr}}} \sum_{|\alpha|=0}^{A_i} \frac{(-1)^{|\alpha|}}{\alpha!} M_i^{[\alpha]} T_{\mathrm{mod}}^{[\alpha]}(\mathbf{R}_i^{\mathrm{MM}}, \mathbf{r}) + \sum_{i=1}^{N_{\mathrm{MM}}^{\mathrm{lr}}} \sum_{|\alpha|=0}^{A_i} \sum_{|\beta|=0}^{B_{\mathrm{QM}}} \frac{(-1)^{|\alpha+\beta|}}{\alpha!\beta!} M_i^{[\alpha]} T^{[\alpha+\beta]}(\mathbf{R}_i^{\mathrm{MM}}, \bar{\mathbf{R}}^{\mathrm{QM}})(\mathbf{r} - \bar{\mathbf{R}}^{\mathrm{QM}})^{\beta} \tag{3}$$

DOI: 10.1021/acs.jctc.9b00424
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

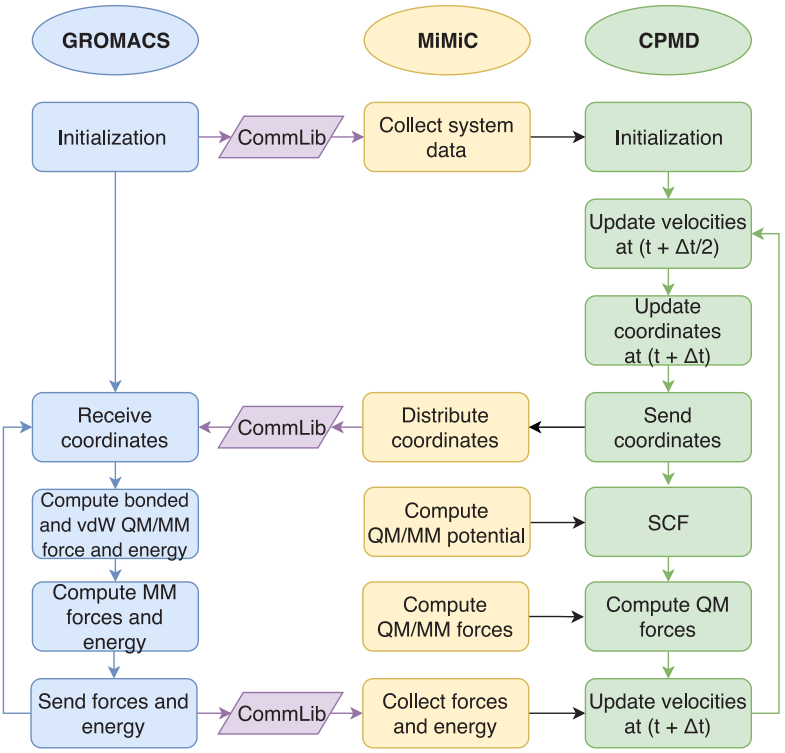


Figure 2. Illustration of the MiMiC-based QM/MM MD workflow using the velocity Verlet algorithm. The workflow shown here is for BO-MD, but a CP-MD workflow would be analogous.

where the first part is the potential from the MM atoms in the short-range domain and the second part is due to the MM atoms in the long-range domain. The computation of both parts of the potential is distributed in a way that is similar to the computation of the short-range part of the electrostatic QM−MM interaction energy but without a reduction within each task group. This means that each MPI process (and associated OpenMP threads) of each group computes its part of the potential and the total potential is straightforwardly obtained via a summation across all task groups.

The electrostatic QM−MM interactions also give rise to additional forces acting on the QM nuclei and MM atoms. The force on a QM atom due to the MM atoms in the short-range domain can be expressed as

$$F_{\mathbf{R}_j^{\mathrm{QM}}}^{\mathrm{es,sr}[\gamma]} = -\sum_{i=1}^{N_{\mathrm{MM}}^{\mathrm{sr}}} \sum_{|\alpha|=0}^{A_i} \frac{(-1)^{|\alpha|}}{\alpha!} M_i^{[\alpha]} T_{\mathrm{mod}}^{[\alpha+\gamma]}(\mathbf{R}_i^{\mathrm{MM}}, \mathbf{R}_j^{\mathrm{QM}}) Z_j^{\mathrm{QM}} \tag{4}$$

with γ being either (1, 0, 0), (0, 1, 0), or (0, 0, 1) representing the Cartesian components of the force. The corresponding force acting on an MM atom in the short-range domain is then

$$F_{\mathbf{R}_i^{\mathrm{MM}}}^{\mathrm{es,sr}[\gamma]} = -\sum_{|\alpha|=0}^{A_i} \frac{(-1)^{|\alpha+\gamma|}}{\alpha!} M_i^{[\alpha]} \left(\int T_{\mathrm{mod}}^{[\alpha+\gamma]}(\mathbf{R}_i^{\mathrm{MM}}, \mathbf{r}) \rho^{\mathrm{QM}}(\mathbf{r}) \mathrm{d}\mathbf{r} + \sum_{j=1}^{N_{\mathrm{QM}}} T_{\mathrm{mod}}^{[\alpha+\gamma]}(\mathbf{R}_i^{\mathrm{MM}}, \mathbf{R}_j^{\mathrm{QM}}) Z_j^{\mathrm{QM}} \right) \tag{5}$$

The forces acting on QM nuclei due to the MM atoms in the long-range domain are obtained via

$$F_{\mathbf{R}_j^{\mathrm{QM}}}^{\mathrm{es,lr}[\gamma]} = -\sum_{i=1}^{N_{\mathrm{MM}}^{\mathrm{lr}}} \sum_{|\alpha|=0}^{A_i} \sum_{|\beta|=1}^{B_{\mathrm{QM}}} \frac{(-1)^{|\alpha+\beta|}}{\alpha!(\beta-\gamma)!} M_i^{[\alpha]} T^{[\alpha+\beta]}(\mathbf{R}_i^{\mathrm{MM}}, \bar{\mathbf{R}}^{\mathrm{QM}}) Z_j^{\mathrm{QM}} \times (\mathbf{R}_j^{\mathrm{QM}} - \bar{\mathbf{R}}^{\mathrm{QM}})^{\beta-\gamma} - \frac{1}{N_{\mathrm{QM}}} \sum_{i=1}^{N_{\mathrm{MM}}^{\mathrm{lr}}} \sum_{|\alpha|=0}^{A_i} \left(\sum_{|\beta|=0}^{B_{\mathrm{QM}}} \frac{(-1)^{|\alpha+\beta|}}{\alpha!\beta!} M_i^{[\alpha]} T^{[\alpha+\beta+\gamma]} \times (\mathbf{R}_i^{\mathrm{MM}}, \bar{\mathbf{R}}^{\mathrm{QM}}) M_{\mathrm{QM}}^{[\beta]}[\rho^{\mathrm{QM}}](\mathbf{R}^{\mathrm{QM}}) + \sum_{|\beta|=1}^{B_{\mathrm{QM}}} \frac{(-1)^{|\alpha+\beta|}}{\alpha!(\beta-\gamma)!} M_i^{[\alpha]} T^{[\alpha+\beta]} \times (\mathbf{R}_i^{\mathrm{MM}}, \bar{\mathbf{R}}^{\mathrm{QM}}) M_{\mathrm{QM}}^{[\beta-\gamma]}[\rho^{\mathrm{QM}}](\mathbf{R}^{\mathrm{QM}}) \right) \tag{6}$$

while the corresponding force on an MM atom is

$$F_{\mathbf{R}_i^{\mathrm{MM}}}^{\mathrm{es,lr}[\gamma]} = -\sum_{|\alpha|=0}^{A_i} \sum_{|\beta|=0}^{B_{\mathrm{QM}}} \frac{(-1)^{|\alpha+\beta+\gamma|}}{\alpha!\beta!} M_i^{[\alpha]} T^{[\alpha+\beta+\gamma]} \times (\mathbf{R}_i^{\mathrm{MM}}, \bar{\mathbf{R}}^{\mathrm{QM}}) M_{\mathrm{QM}}^{[\beta]}[\rho^{\mathrm{QM}}](\mathbf{R}^{\mathrm{QM}}) \tag{7}$$

The parallel computation of the forces is performed analogously to the computation of the energy. First, each MPI process (and associated OpenMP threads) within a group computes a partial contribution of force acting on each of the atoms belonging to the group. Then, a global summation of partial forces is performed in order to obtain the total force acting on each atom of the system.

SIMULATION WORKFLOW

The workflow of a MiMiC-based QM/MM BO-MD simulation using CPMD and GROMACS is illustrated in Figure 2 (an analogous workflow would apply for CP-MD). Both CPMD and GROMACS run simultaneously. After initialization, MiMiC collects system data from GROMACS (number of atoms, atom types, coordinates, etc.) and propagates it to CPMD. Once in the MD loop, the first two steps of the velocity

DOI: 10.1021/acs.jctc.9b00424
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

Verlet integrator, half time step of velocity and a complete time step of position updates, are performed as usual. After the coordinates have been updated, they are distributed to GROMACS through MiMiC using a communication library (CommLib) (see the Interprogram Communication section for further details). Then, both CPMD and GROMACS proceed computing energies and forces of the QM and MM subsystems, respectively, concurrently. In the case of BO-MD, as in the example shown in Figure 2, this includes a self-consistent field (SCF) procedure to solve the KS equations in order to determine the electronic density. This requires also the electrostatic potential from the MM part (eq 3), which is computed by MiMiC and added to the KS-DFT external potential in CPMD. After convergence of the SCF procedure, CPMD computes the QM forces and receives the QM/MM part of the forces (eqs 4−7) from MiMiC. Finally, MiMiC collects the energy and forces from GROMACS and propagates them to CPMD, which then performs the second velocity update and goes back to the beginning of the MD loop.

In the present work, the equations of motion are integrated by CPMD, including those of the MM atoms. This implies that bond, angle, and other constraints among MM atoms must be enforced by CPMD. The (serial) RATTLE algorithm[14] implemented in CPMD as the standard constraint solver is suitable for systems of up to a few hundred atoms which is typical of pure QM systems but may become a computational bottleneck for larger systems (hundreds of thousands of atoms are not uncommon in MM). To overcome this problem, we implemented an improved version of RATTLE proposed by Weinbach and Elber.[15] This improved version takes advantage of sparse linear algebra to impose the constraints, which greatly reduces the associated computational cost. Even though the method is in principle fully MPI parallelizable, here we implement a version relying only on OpenMP multithreading. The threading is applied to the sparse linear algebra operations by distributing rows of the sparse matrix between threads. MPI and hybrid MPI/OpenMP versions of the method will be implemented in the future to further enhance the scalability.

INTERPROGRAM COMMUNICATION

The loose-coupling model used in the MiMiC framework requires communication of data, e.g., atomic positions and forces, between programs which are running independently. To achieve this objective MiMiC provides a communication library (CommLib), which has been designed to minimize the coding effort in the external programs interfaced by the MiMiC framework. CommLib does not interfere with pre-existing communication layers within the coupled programs, e.g., their MPI parallelization scheme does not need to be altered. In practice, one simply invokes the send and/or receive routines in CommLib to exchange data between two programs using an ID associated with a specific domain of the multiscale representation (e.g., QM, MM, etc.) which is created during CommLib initialization.

To avoid interference with MPI communication layers that may be present in the coupled programs, CommLib uses client-server *intercommunication* which is an MPI2 feature. This allows direct exchange of data between processes belonging to different communicators, i.e., the `COMM_WORLD` communicators of the independent programs. Moreover, we exploit the features of the communication fabric (e.g., Infiniband or Intel Omni-Path architectures) available on the system to reduce the communication overhead. Our benchmarks show that the overhead introduced by CommLib is negligible or minor compared to the wall time even for large systems (see the Performance Benchmarks section). CommLib can be extended to use other communication mechanisms in cases where MPI2 is not available without affecting its use in the coupled programs.

COMPUTATIONAL DETAILS

The performance of our MiMiC-based QM/MM implementation was tested in a number of benchmarks. These report the speedup (i.e., $S_N = \frac{t_1}{t_N}$ where N is the number of compute nodes, and t_1 is the timing of one node), parallel efficiency ($\mathrm{PE} = \frac{S_N}{N}*100\%$), and absolute timing per MD step. Here we define the scaling limit as the number of cores at which the parallel efficiency goes below 70% threshold. The systems differ by their size and complexity (see below). To benchmark MiMiC, we consider two exchange−correlation (XC) functionals, BLYP,[16,17] to test the performance with a typical generalized-gradient approximation (GGA) functional, and B3LYP,[17−21] to test a hybrid one. The former scales linearly with system size, whereas the latter scales quadratically. It has been shown that the implementation of hybrid functionals in CPMD can efficiently scale to thousands of cores.[12] Therefore, B3LYP-based benchmarks allows us to test the scalability of MiMiC in very challenging parallel conditions.

Benchmarks were performed on clusters hosted by the Jülich Supercomputing Centre and RWTH Aachen, namely JURECA,[22] JUWELS,[23] and CLAIX-2018.[24] All three clusters have a fat-tree topology. JURECA's compute nodes are equipped with two Intel Xeon E5-2680 12-core CPUs, while JUWELS has two 24-core Intel Xeon Platinum 8168 CPUs per node. The nodes in CLAIX-2018 have two 24-core Intel Xeon Platinum 8160 CPUs.

The set of benchmark systems consist of two solute−solvent systems and one protein system. They were chosen to represent a diverse composition and broad range of sizes for the QM and MM systems making it possible to explore the scaling behavior of several combinations of QM and MM systems. The solute−solvent systems include *n*-butanol solvated in acetone solution (Figure 3) and *N*-(6-methoxyquinolyl) acetoethyl ester (MQAE) in a water solution (Figure 4). The protein system is a chloride ion channel (ClC-ec1) embedded in a lipid bilayer[25] (Figure 5). The initial structure for butanol in acetone was taken from ref 4. Both BO-MD and CP-MD were investigated in the benchmarks. We used a time step of 20 au (∼0.48 fs) for BO-MD and 5 au (∼0.12 fs) for CP-MD. Scaling of the MiMiC-based QM/MM is reported with respect to the amount of cores allocated for CPMD. In the present benchmarks it was not necessary to allocate more than one compute node to GROMACS.

The butanol in acetone system consists of one butanol (QM) and 13,948 acetone molecules (MM), totalling 139,495 atoms, which thus represents a system with a *small* QM subsystem and a *large* MM subsystem. The 16 orbitals of butanol were expanded on a real-space FFT mesh of 240^3 points corresponding to a plane wave cutoff of 85 Ry. The number of atoms in the solvated MQAE system is 16,396 of which either 34 (MQAE) or 151 (MQAE + 39 × H_2O) are in the QM subsystem, representing systems with a *small* and a *large* QM subsystem, respectively, together with a *small* MM subsystem. At a cutoff of 80 Ry, the 47 orbitals of the small QM subsystem

DOI: 10.1021/acs.jctc.9b00424
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

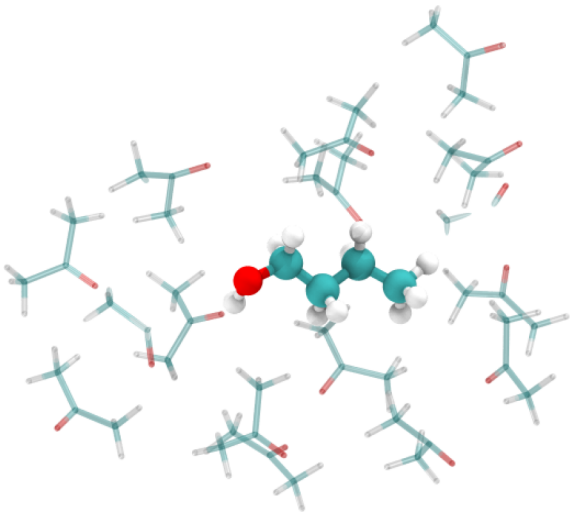


Figure 3. Butanol molecule solvated in acetone. Only part of the total system is shown. The QM subsystem is displayed in a ball-and-stick representation.

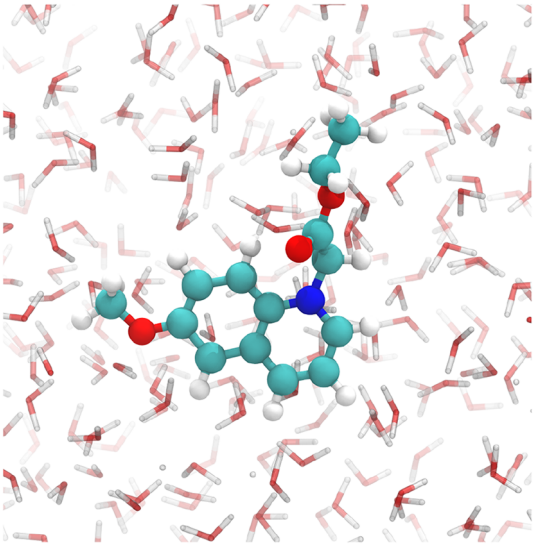


Figure 4. MQAE in water solution. Only part of the total system is shown. The QM subsystem is displayed in a ball-and-stick representation.

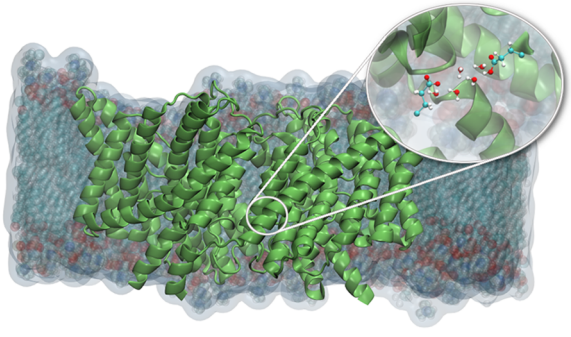


Figure 5. ClC-ec1 ion channel in a lipid bilayer. Water solvent is not shown. The QM subsystem is displayed in the inset in a ball-and-stick representation.

are expanded on a mesh of 216^3 points, whereas the 203 orbitals of the large QM subsystem are expanded on a 288^3 mesh. The protein systems considered here also vary in size. The ClC-ec1 system consists of 150,925 atoms, where either 19 or 295 atoms were selected to represent a *small* and *large* QM subsystem, respectively, combined with a *large* MM subsystem. The real-space mesh for the small and large QM subsystems consist of 240^3 and $360 \times 300 \times 300$, respectively, corresponding to a 90 Ry cutoff. The amount of orbitals totalled 49 in the small QM subsystem and 383 in the large QM subsystem. Apart from the MQAE and ClC-ec1 setups with large QM regions, which were only treated at the BLYP level, benchmarks for all other systems were performed using the GGA XC functional BLYP as well as the hybrid-GGA XC functional B3LYP. All simulations were performed using Troullier−Martins norm-conserving pseudopotentials[26] using the new XC driver by Bircher and Rothlisberger.[27] Note that the setups detailed above do not necessarily correspond to a realistic setup suitable for a production simulation but are prepared in order to demonstrate the performance of systems with different QM and MM subsystem sizes. The timings are given as averages over 5 steps in BO-MD and 10 steps in CP-MD. However, in the case of BO-MD using B3LYP, only 3 time steps were used due to the large amount of time needed when running on a single node.

As already discussed in the Parallelization Schemes section, both MiMiC and CPMD implement several parallelization strategies: parallelization over FFT planes, task groups, and multithreading. These complementary approaches are exploited in our benchmarks to obtain good performance, as discussed for each system in their corresponding section below.

PERFORMANCE BENCHMARKS

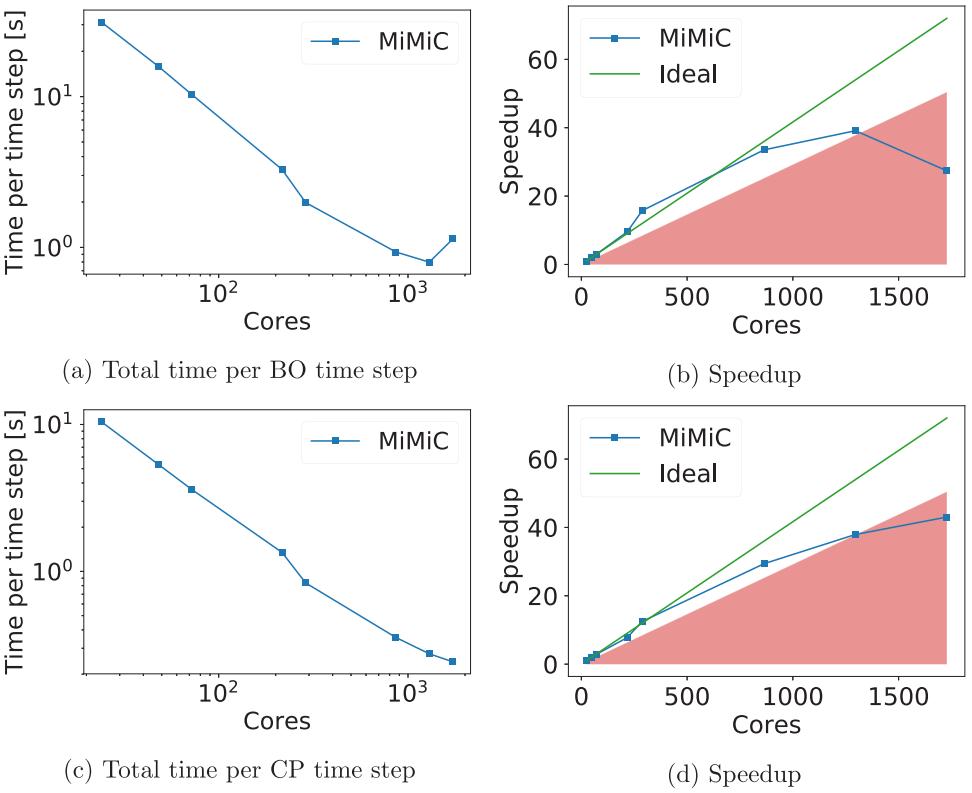
In this section, we present benchmarks of the MiMiC-based QM/MM simulations of the systems described in the previous section.

MQAE. The MQAE system is relatively small. In order to benchmark the performance on systems with small and large QM subsystems combined with a small MM subsystem, we ran two sets of benchmarks: in the first one we included only the solute in the QM part, and in the second one we also added solvent molecules that were within 3.0 Å of the solute.

Benchmarks using the small QM system with BLYP show that MiMiC retains high parallel efficiency up to 1,296 cores within BO-MD with a total time per time step equal to approximately 0.8 s (see Figure 6a). With four OpenMP threads per MPI process the speedup flattens at 864 cores. A higher scalability is achieved with six threads per process. Using task groups does not extend the scalability any further (last point in Figure 6b). CP-MD behaves in a similar way allowing the efficient use of up to 1,296 cores (0.28 s per time step).

It is noticeable that in the BO benchmark (Figure 6b), we observed superlinear scaling around 300 cores. The reason for such a peculiar behavior is that at moderate numbers of cores the use of cache memories improves with the number of cores.[28] Let us illustrate this phenomenon with an example. The JURECA CPUs have 30 MB L3 (level 3) cache shared across all cores in addition to 32 KB and 256 KB of L1 (level 1) and L2 (level 2) caches, respectively, on each core. The size of the FFT mesh for MQAE is estimated to be roughly 80 MB. With two CPUs on a single node, the grid is split between those into 40 MB portions. This means that the portion available for a single CPU does not fit into the cache available on said chip. However, as the number of CPUs increases, the portion sizes

DOI: 10.1021/acs.jctc.9b00424
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX



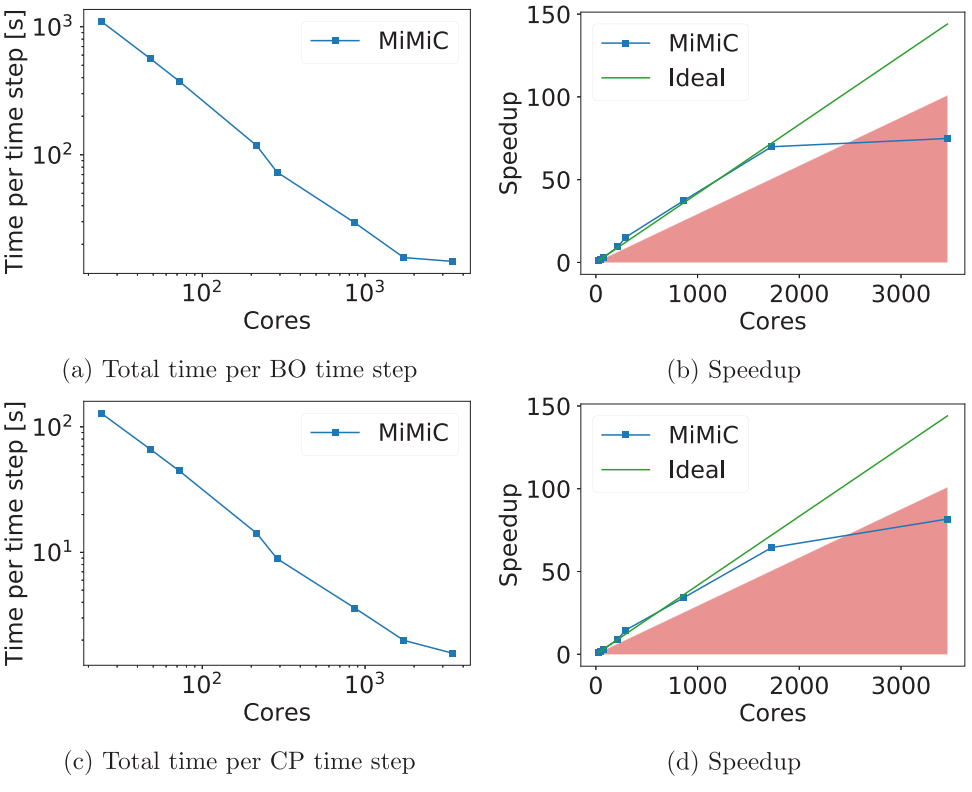
(a) Total time per BO time step

(b) Speedup

(c) Total time per CP time step

(d) Speedup

Figure 6. Scaling performance of the MQAE system with the small QM subsystem both within BO-MD and CP-MD using BLYP. Simulations were run on JURECA. Red area shows the 70% parallel efficiency threshold.



(a) Total time per BO time step

(b) Speedup

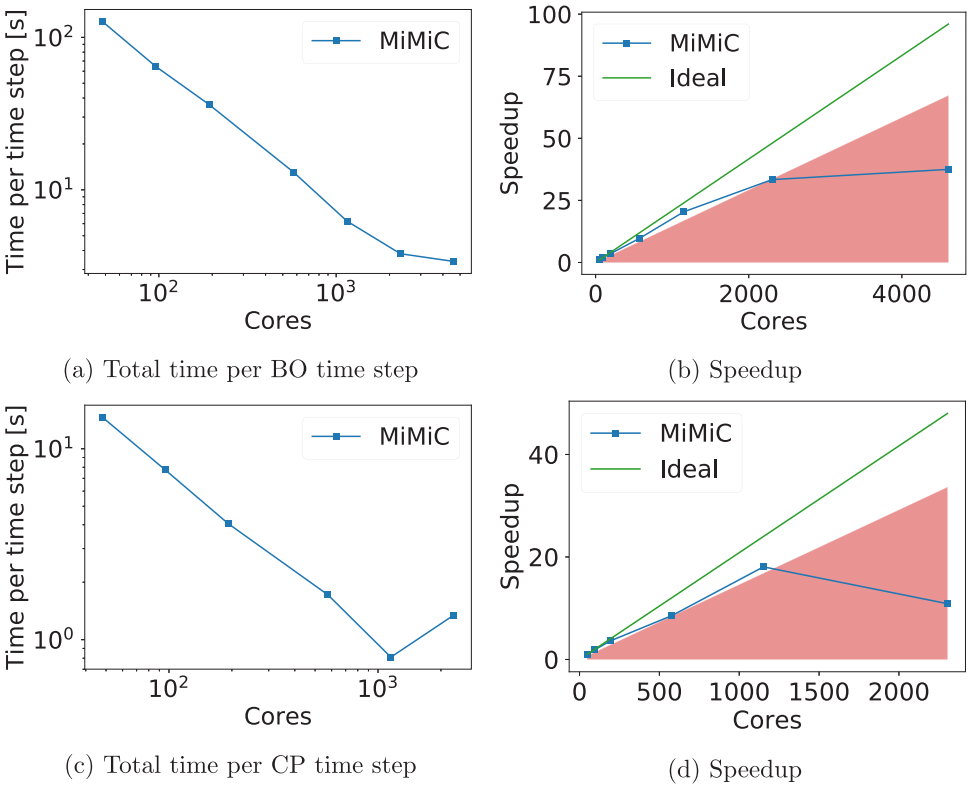
(c) Total time per CP time step

(d) Speedup

Figure 7. Scaling performance of the MQAE with the small QM subsystem both within BO-MD and CP-MD using B3LYP. Simulations were run on JURECA. Red area shows the 70% parallel efficiency threshold.

per CPU decrease. First, portions will reach a size where they fit into the L3 cache and, once the grid is decomposed even further, the part of the grid treated by a single core will ultimately fit into the L2 cache.

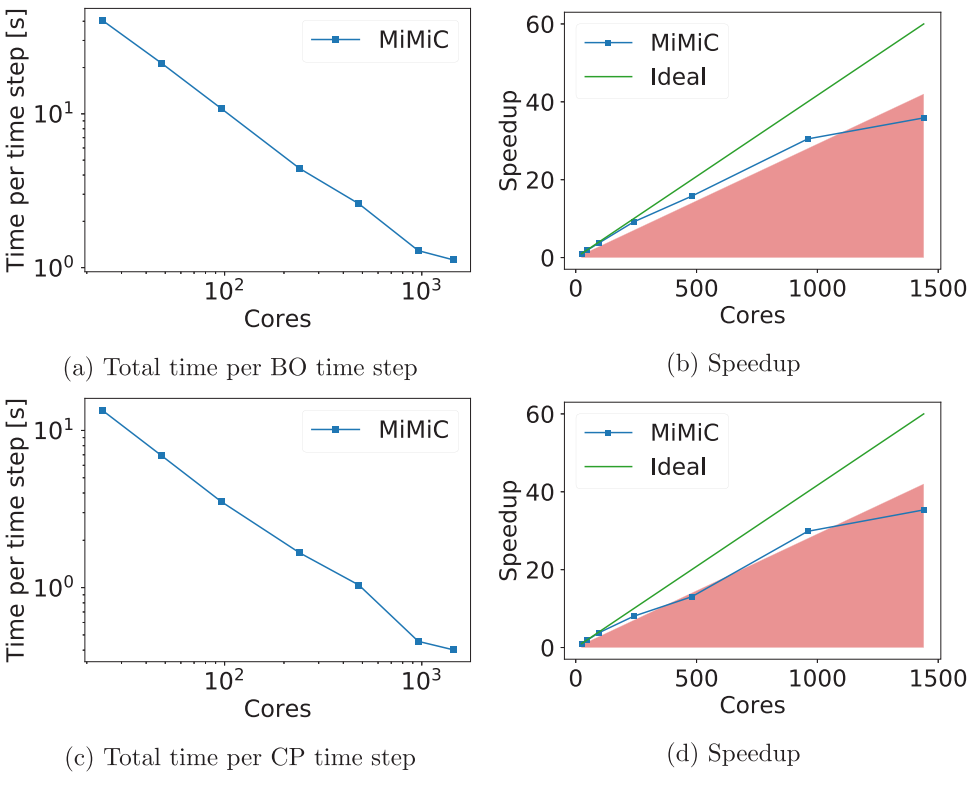
DOI: 10.1021/acs.jctc.9b00424
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX



(a) Total time per BO time step (b) Speedup

(c) Total time per CP time step (d) Speedup

Figure 8. Scaling performance of the MQAE system with the large QM subsystem both within BO-MD and CP-MD using BLYP. Simulations were run on CLAIX-2018. Red area shows the 70% parallel efficiency threshold.



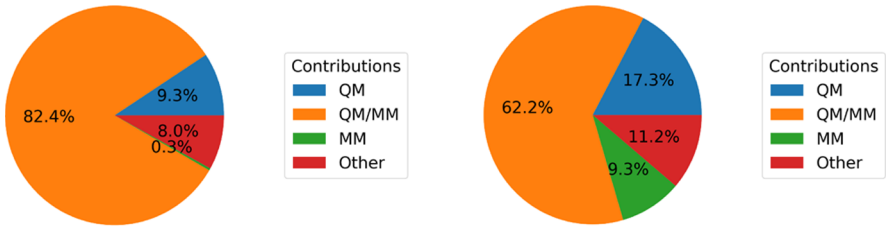
(a) Total time per BO time step (b) Speedup

(c) Total time per CP time step (d) Speedup

Figure 9. Scaling performance of the butanol system both within BO-MD and CP-MD using BLYP. Simulations were run on JURECA. Red area shows the 70% parallel efficiency threshold.

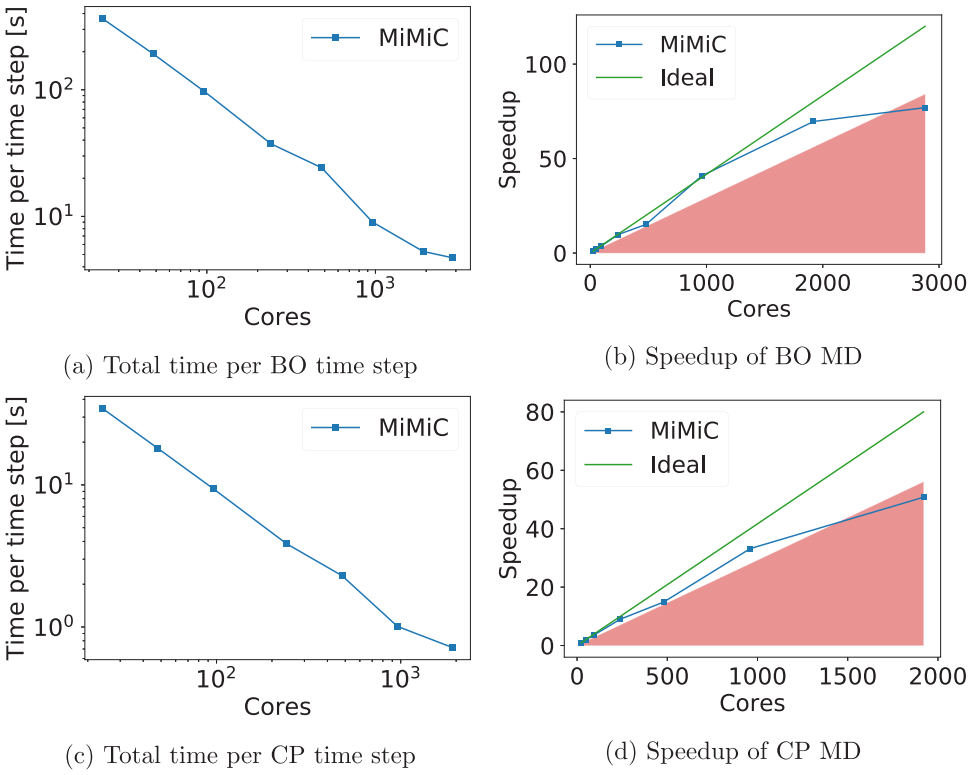
The benchmarks based on the B3LYP XC functional show that the use of two task groups extends the scaling to 1,728 cores (see Figure 7). The scaling is limited by the fact that the number of orbitals in this system is a relatively small prime number (47), which inevitably leads to load imbalance of the exact exchange computation that, in turn, constitutes the

DOI: 10.1021/acs.jctc.9b00424
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX



(a) Using 24 cores (b) Using 960 cores

Figure 10. Breakdown of time spent on one time step of CP-MD using BLYP for butanol. Slice "Other" includes the integrator with auxiliary functions (e.g., computation of observables).



(a) Total time per BO time step (b) Speedup of BO MD

(c) Total time per CP time step (d) Speedup of CP MD

Figure 11. Scaling performance of the butanol system both within BO-MD and CP-MD using B3LYP. Simulations were run on JURECA. Red area shows the 70% parallel efficiency threshold.

bottleneck of the calculation both with respect to execution time and parallelization. Finally, enforcing the orbital orthogonality at each CP-MD time step, even though the procedure is parallelized, may further hamper the speedup. Overall, the best achieved performance for B3LYP simulations was 15.88 s per time step of BO-MD and 2.00 s per time step for CP-MD. We remark that the factors limiting the scalability are intrinsic to the QM calculation rather than QM/MM and MiMiC.

Based on the larger dimension of the grid (288^3), it can be expected that simulations using the large QM subsystem will have a scaling limit at least 1.3 times higher than the small QM case. However, as seen in Figure 8, the BLYP-based BO-MD simulation can use over 2,300 cores by employing two task groups (which is roughly 1.8 times more compared to the scaling limit of the smaller system), thus bringing the total time per time step down to about 3.8 s. The scaling performance of CP-MD, on the other hand, is as expected, allowing efficient use of 1.3 times more cores (1,152) with a total time per time step of 0.81 s. This difference between BO-MD and CP-MD can be due to multiple factors. First, the serial parts of the integrator, including the constraint solver (which is only parallelized using threading), stall the performance in the subsecond range. Then, the use of task groups adds more communication in CP-MD where it is associated with the computation of the orbital overlap between groups; this substantial communication step is absent in GGA-only BO-MD. These factors taken together explain the superior scaling of the BO-MD simulation with respect to its CP counterpart.

Butanol. This system has a very small QM subsystem (15 atoms) combined with a large MM subsystem (139,480 atoms). Therefore, it is expected that the QM/MM part plays a major role performance-wise.

Results of BLYP-based CP-MD benchmarks are shown in Figure 9c. As expected, QM/MM contributions are especially prominent here, where they take up to 82% of the time on a single node, as shown in Figure 10a. Thanks to the efficient parallelization and excellent scaling of the QM/MM contributions computed by MiMiC, the time per time step of CP-MD can be brought down to approximately 0.45 s using 960 cores

DOI: 10.1021/acs.jctc.9b00424
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

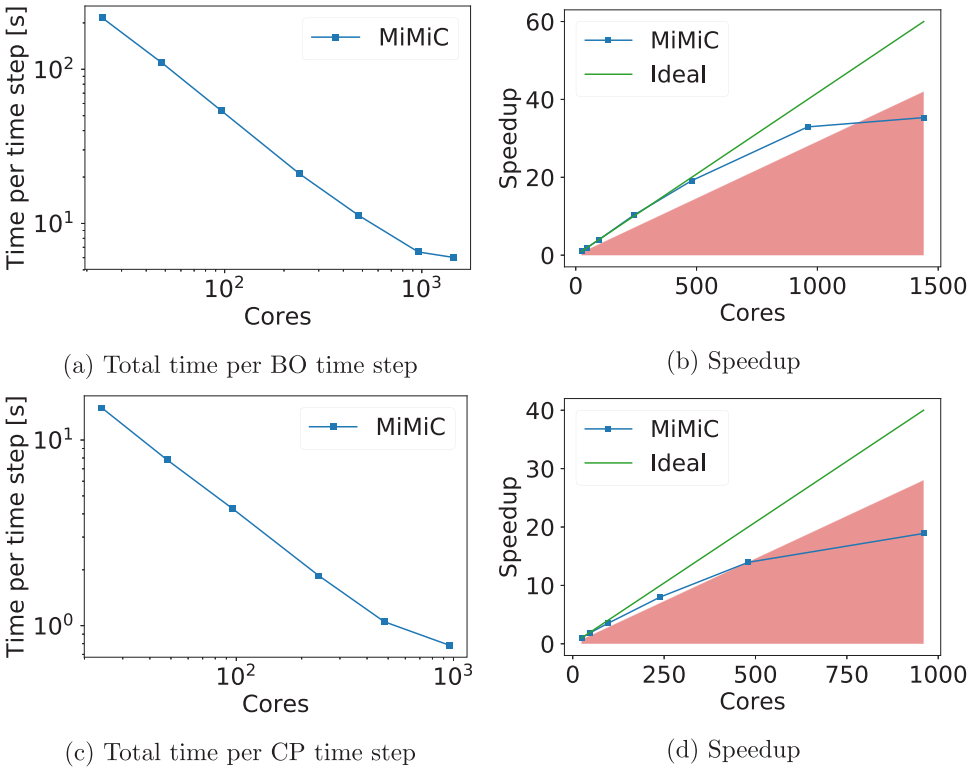


Figure 12. Scaling performance of the ClC-ec1 system with the small QM subsystem both within BO-MD and CP-MD using BLYP. Simulations were run on JURECA. Red area shows the 70% parallel efficiency threshold.

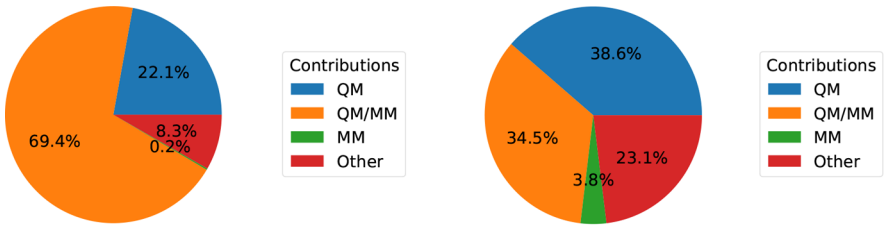


Figure 13. Breakdown of time spent on one time step of CP-MD using BLYP for ClC-ec1 with the small QM subsystem. Slice "Other" includes the integrator with auxiliary functions (e.g., computation of observables).

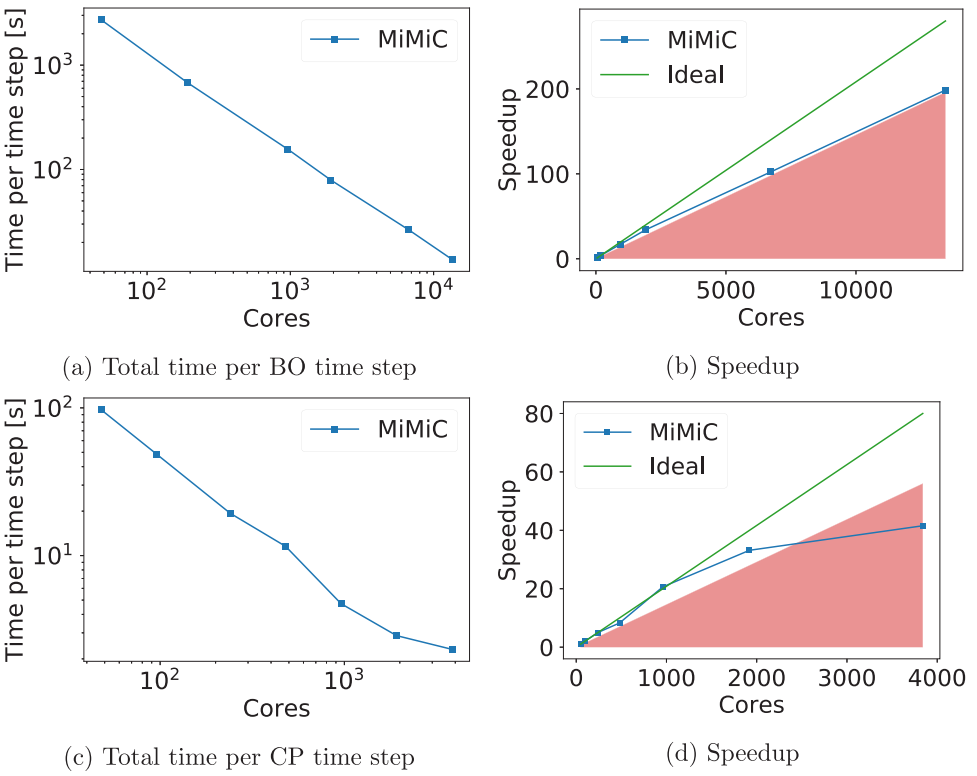
(see Figure 9c). In addition, as mentioned above, further scalability is prevented by the small size of the FFT grid (240^3) and the small number of orbitals (16), which limits the benefits of task groups in CPMD. Another aspect limiting the scalability to a lesser degree is the communication between CPMD and GROMACS, which in the present system is not negligible. These bottlenecks appear mostly due to the small size of the QM part compared to the substantial extent of the MM subsystem. The detailed breakdown of the time spent computing each of the contributions at the scaling limit is shown in Figure 10b.

BLYP-based BO-MD exhibits a similar scaling behavior as the BLYP-based CP-MD and scales up to the same limit of 960 cores (Figure 9a), indicating that the scaling differences in the QM part are minor due to the dominant effect of the QM/MM contributions. Again, the efficient parallelization scheme allows us to substantially reduce the time needed for a BO-MD step (1.29 s). Further scaling is mostly hampered by the bottlenecks summarized for CP-MD above.

In the B3LYP-based simulations, the use of two task groups extends the scaling limit of a BO-MD by a factor of 2 (Figure 11). This makes it possible to use up to 1,920 cores with an efficiency of 85% and a time per time step of about 5.3 s only. Further addition of task groups, though, causes a decline in performance, typical of a system with a small number of orbitals where the additional communication between groups in the QM part outweighs the gain in performance due to task distribution. Similarly, for the B3LYP-based CP-MD run (time per time step of 1.01 s), the application of task grouping does not improve the scaling limit, which is the same as the one observed for BLYP. Upon investigation, we found that the reasons for the decline in performance are of the same nature as the stalling of the scaling performance of BLYP-based CP-MD outlined above (i.e., the serial integrator and the communication overhead).

ClC-ec1. The ClC-ec1 system has the largest number of MM atoms considered here (150,925). Therefore, QM/MM interactions are of paramount importance for simulations of this system. In order to examine the effect of the relative size of

DOI: 10.1021/acs.jctc.9b00424
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX



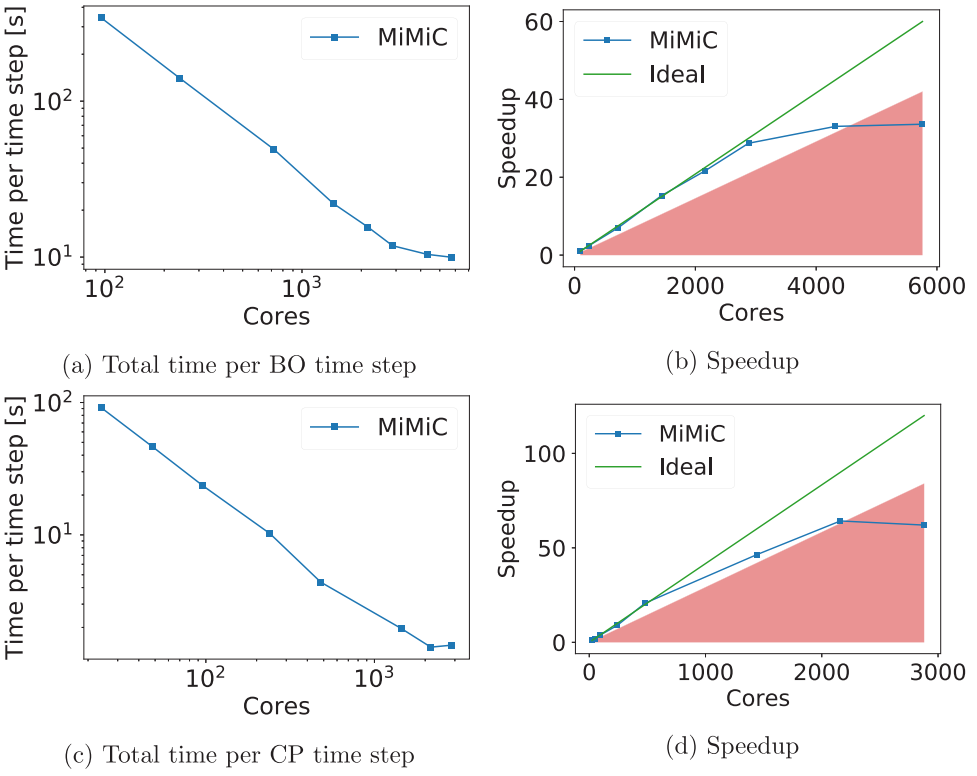
(a) Total time per BO time step

(b) Speedup

(c) Total time per CP time step

(d) Speedup

Figure 14. Scaling performance of the ClC-ec1 system with the small QM subsystem both within BO-MD and CP-MD using B3LYP. Simulations were run on JUWELS. Red area shows the 70% parallel efficiency threshold. The first point of the graphs in (a) and (b) is extrapolated because of excessive computational cost of a single time step.



(a) Total time per BO time step

(b) Speedup

(c) Total time per CP time step

(d) Speedup

Figure 15. Scaling performance of the ClC-ec1 system with the large QM subsystem both within BO-MD and CP-MD using BLYP. Simulations were run on JURECA. Red area shows the 70% parallel efficiency threshold. The reference for BO-MD is four nodes instead of one due to the limited amount of memory available on a single node.

DOI: 10.1021/acs.jctc.9b00424
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

the QM and MM subsystems on the performance, we considered two QM subsystems of different size containing 19 and 295 atoms. The latter is obtained from the former by including all residues that are within 3 Å of the atoms of the former.

The behavior of the BLYP-based simulations with a small QM system is similar to the one of butanol (see Figure 12). BO-MD scales up to 960 cores with a single time step taking 6.53 s. On the other hand, scalability is poorer for CP-MD. For this special case, it is not possible to use more than 480 cores with reasonable efficiency. The reason for this lies in the large size of the MM subsystem and the associated large number of constraints. This, together with the serial velocity- and coordinate-updates in the CPMD integrator, stalls the scaling (see Figure 13).

The picture is rather different when the small QM subsystem is treated at the B3LYP level. These benchmarks have been run on the JUWELS supercomputer, due to the significant computational resources required. The use of task groups makes it possible to push the scaling limit up to 13,440 cores by using 14 groups, thus reducing the wall time spent per each BO-MD time step to around 13.76 s (Figure 14). In line with the scaling behavior observed for the other systems, the use of task groups is less beneficial for CP-MD. Even so, it is possible to use two task groups, thus pushing the scaling limit to 1,920 cores spending 2.86 s per each CP time step. The stalling of the scaling can again be attributed to the inferior scaling of the orthogonalizer used to enforce the orthonormality constraint in CP dynamics at every time step, as well as the number of orbitals (49) where, with its large prime factors, it is not trivial to achieve a reasonable load distribution.

By considering a setup with an enlarged QM subsystem, the performance of the MiMiC-based QM/MM can be benchmarked under conditions where both QM and MM subsystems are large. This system also requires use of a noncubic QM box. Since the QM mesh has 1.5 times more points along one dimension, an improvement in the scaling limit is expected due to better scaling in the low-level grid parallelization. The results for this setup can be found in Figure 15. Indeed, BLYP-based BO-MD scales excellently, up to 4,320 cores with an efficiency above 70% (rather than 980 cores as for the smaller QM subsystem). The scaling of CP-MD also shows an impressive improvement, extending the scaling limit observed for the small QM subsystem by as much as a factor of 4.5. The cost of force computations, which are highly parallelized, becomes more relevant compared to the orthogonalization of orbitals and the constraint solver, which both on their own have shown inferior scaling with respect to the computation of forces. Additional benefits come from the efficient use of threading. Here it is possible to increase the number of OpenMP threads per MPI process from four to six together with two task groups for BO-MD. In the CP-MD simulations, up to six threads per MPI process were used, but yet again, task groups did not offer any performance improvement. Therefore, the best possible timings obtained for BO-MD and CP-MD with BLYP are 10.43 and 1.42 s per time step, respectively.

SUMMARY

In this work, we have presented parallelization strategies and extensive performance benchmarks for the MiMiC-based QM/MM implementation[4] interfacing CPMD and GROMACS as the QM and MM engines, respectively. This work demonstrates for the first time that flexibility can be achieved without compromising efficiency. Indeed, our loosely coupled MPMD approach allowed us to achieve extreme parallel performance by combining an efficient parallelization of the QM/MM contributions with the highly parallel CPMD and GROMACS software packages, both of which can run concurrently and use their own optimal parallelization strategies.

We demonstrate efficient utilization of several thousand cores bringing the average time per time step down to the few-seconds range for BO-MD and subsecond range for CP-MD. A major improvement of the parallel algorithm is achieved via combined use of a MPI-based task group approach and OpenMP multithreading. This paves the way for QM/MM simulations in the time scale of nanoseconds within the duration of a typical computational project even when using hybrid XC functionals and enables investigations of large and complex (bio)chemical systems within plane-wave/pseudopotential DFT-based QM/MM.

AUTHOR INFORMATION

Corresponding Authors

*E-mail: v.bolnykh@fz-juelich.de.
*E-mail: jogvan.m.olsen@uit.no.
*E-mail: p.carloni@fz-juelich.de.
*E-mail: ursula.roethlisberger@epfl.ch.

ORCID

Viacheslav Bolnykh: 0000-0002-7090-1993
Jógvan Magnus Haugaard Olsen: 0000-0001-7487-944X
Simone Meloni: 0000-0002-3925-3799
Martin P. Bircher: 0000-0002-6905-3130
Emiliano Ippoliti: 0000-0001-5513-8056
Paolo Carloni: 0000-0002-9010-0149
Ursula Rothlisberger: 0000-0002-1704-8591

Author Contributions

Conceptualization: J.M.H.O. (equal), S.M. (equal), and U.R. (equal); Software: V.B. (equal), J.M.H.O. (equal), S.M. (supporting), and M.P.B. (supporting); Investigation: V.B. (lead); Resources: P.C. (equal) and E.I. (equal); Visualization: V.B. (lead); Writing − Original Draft: V.B. (lead), J.M.H.O. (supporting), S.M. (supporting), M.P.B. (supporting), and E.I. (supporting); Writing − Review and Editing: V.B. (equal), J.M.H.O. (equal), S.M. (equal), E.I. (equal), M.P.B. (equal), P.C. (equal), and U.R. (equal); Supervision: J.M.H.O. (equal), U.R. (equal), and P.C (equal); Funding Acquisition: J.M.H.O. (equal), P.C. (equal), and U.R. (equal).

Funding

V.B. and P.C. acknowledge funding received within the European Union's Horizon 2020 program from project HPC-LEAP (MSCA-ITN-2014-EJD, Grant ID: 642069). J.M.H.O. acknowledges financial support from the Danish Council for Independent Research (DFF) through the Sapere Aude research career program (Grant ID: DFF-1325-00091 and DFF-1323-00744), the Carlsberg Foundation (Grant ID: CF15-0823), and the Research Council of Norway through its Centres of Excellence scheme (Project ID: 262695). E.I. and P.C. acknowledge funding received within the European Union's Horizon 2020 program from the BioExcel Center of Excellence (EINFRA-5-2015, Grant ID: 675728). P.C. also acknowledges the funding by the Deutsche Forschungsgemeinschaft via FOR 2518 DynIon project P6. U.R. acknowledges funding from the Swiss National Science Foundation via the NCCR MUST and individual grants. Computational resources

DOI: 10.1021/acs.jctc.9b00424
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

were provided by Jülich Supercomputing Center and RWTH Aachen University.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors thank IBM Zürich Research Laboratory and especially Teodoro Laino, Valery Weber, and Alessandro Curioni for their help with the development of the interface in CPMD. The authors also thank Mark James Abraham, Berk Hess, and Erik Lindahl for valuable advice concerning the GROMACS interface. The authors are grateful to Maria Gabriella Chiarello for providing the pre-equilibrated structure of ClC-ec1.

■ REFERENCES

(1) Warshel, A.; Levitt, M. Theoretical studies of enzymic reactions: Dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme. *J. Mol. Biol.* **1976**, *103*, 227−249.

(2) Senn, H. M.; Thiel, W. QM/MM Methods for Biomolecular Systems. *Angew. Chem., Int. Ed.* **2009**, *48*, 1198−1229.

(3) Brunk, E.; Rothlisberger, U. Mixed Quantum Mechanical/Molecular Mechanical Molecular Dynamics Simulations of Biological Systems in Ground and Electronically Excited States. *Chem. Rev.* **2015**, *115*, 6217−6263. PMID: 25880693.

(4) Olsen, J. M. H.; Bolnykh, V.; Meloni, S.; Ippoliti, E.; Bircher, M. P.; Carloni, P.; Rothlisberger, U. MiMiC: A Novel Framework for Multiscale Modeling in Computational Chemistry. *J. Chem. Theory Comput.* **2019**, *15*, 3810.

(5) Hutter, J.; Alavi, A.; Deutsch, T.; Bernasconi, M.; Goedecker, S.; Marx, D.; Tuckerman, M.; Parrinello, M. CPMD Copyright IBM Corp 1990−2019, MPI für Festkörperforschung, Stuttgart, 1997−2012. http://www.cpmd.org/.

(6) Hess, B.; Kutzner, C.; van der Spoel, D.; Lindahl, E. GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation. *J. Chem. Theory Comput.* **2008**, *4*, 435−447.

(7) Abraham, M. J.; Murtola, T.; Schulz, R.; Pall, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **2015**, *1-2*, 19−25.

(8) Laio, A.; VandeVondele, J.; Rothlisberger, U. Hamiltonian electrostatic coupling scheme for hybrid Car-Parrinello molecular dynamics simulations. *J. Chem. Phys.* **2002**, *116*, 6941−6947.

(9) Marx, D.; Hutter, J. *Ab initio molecular dynamics: basic theory and advanced methods*; Cambridge University Press: 2009; DOI: 10.1017/CBO9780511609633.

(10) Wiggs, J.; Jønsson, H. A hybrid decomposition parallel implementation of the Car-Parrinello method. *Comput. Phys. Commun.* **1995**, *87*, 319−340.

(11) Hutter, J.; Curioni, A. Car−Parrinello molecular dynamics on massively parallel computers. *ChemPhysChem* **2005**, *6*, 1788−1793.

(12) Weber, V.; Bekas, C.; Laino, T.; Curioni, A.; Bertsch, A.; Futral, S. *Shedding Light on Lithium/Air Batteries Using Millions of Threads on the BG/Q Supercomputer*. 2014 IEEE 28th International Parallel and Distributed Processing Symposium. 2014; DOI: 10.1109/ipdps.2014.81.

(13) Hutter, J.; Curioni, A. Dual-level parallelism for ab initio molecular dynamics: Reaching tera op performance with the CPMD code. *Parallel Computing* **2005**, *31*, 1−17.

(14) Andersen, H. C. Rattle: A velocity version of the shake algorithm for molecular dynamics calculations. *J. Comput. Phys.* **1983**, *52*, 24−34.

(15) Weinbach, Y.; Elber, R. Revisiting and parallelizing SHAKE. *J. Comput. Phys.* **2005**, *209*, 193−206.

(16) Becke, A. D. Density-functional exchange-energy approximation with correct asymptotic behavior. *Phys. Rev. A: At., Mol., Opt. Phys.* **1988**, *38*, 3098−3100.

(17) Lee, C.; Yang, W.; Parr, R. G. Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1988**, *37*, 785−789.

(18) Becke, A. D. A new mixing of Hartree-Fock and local density-functional theories. *J. Chem. Phys.* **1993**, *98*, 1372−1377.

(19) Becke, A. D. Densityfunctional thermochemistry. III. The role of exact exchange. *J. Chem. Phys.* **1993**, *98*, 5648−5652.

(20) Vosko, S. H.; Wilk, L.; Nusair, M. Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis. *Can. J. Phys.* **1980**, *58*, 1200−1211.

(21) Stephens, P. J.; Devlin, F. J.; Chabalowski, C. F.; Frisch, M. J. Ab Initio Calculation of Vibrational Absorption and Circular Dichroism Spectra Using Density Functional Force Fields. *J. Phys. Chem.* **1994**, *98*, 11623−11627.

(22) JURECA Compute cluster. http://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JURECA/Configuration/Configuration_node.html (accessed 2019-04-19).

(23) JUWELS Compute cluster. http://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JUWELS/Configuration/Configuration_node.html (accessed 2019-04-19).

(24) CLAIX-2018 Compute cluster. https://doc.itc.rwth-aachen.de/display/CC/Hardware+of+the+RWTH+Compute+Cluster (accessed 2019-04-19.

(25) Dutzler, R.; Campbell, E. B.; Cadene, M.; Chait, B. T.; MacKinnon, R. X-ray structure of a ClC chloride channel at 3.0Å reveals the molecular basis of anion selectivity. *Nature* **2002**, *415*, 287−294.

(26) Troullier, N.; Martins, J. L. Efficient pseudopotentials for plane-wave calculations. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1991**, *43*, 1993−2006.

(27) Bircher, M. P.; Rothlisberger, U. Plane-Wave Implementation and Performance of -la- Carte Coulomb-Attenuated Exchange-Correlation Functionals for Predicting Optical Excitation Energies in Some Notorious Cases. *J. Chem. Theory Comput.* **2018**, *14*, 3184−3195.

(28) Konôpka, M.; Turanský, R.; Doltsinis, N. L.; Marx, D.; Štich, I. *High Performance Computing in Science and Engineering '08*; Springer Berlin Heidelberg: pp 95−108, DOI: 10.1007/978-3-540-88303-6_8.

DOI: 10.1021/acs.jctc.9b00424
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX